CLAIMS

I claim:

1. A method for computing the pitch names (i.e., C♯4, B♭3, etc.) of notes in a representation of music in which at least the onset time and MIDI note number (or chromatic pitch) of each note is given or can be trivially computed, comprising the steps of

   (a) computing for each pitch class $0 \leq p \leq 11$ and each note n in the input, the pitch letter name $S(p, n) \in \{A, B, C, D, E, F, G\}$, that n would have if p were the tonic at the point in the piece where n occurs (assuming that the notes are spelt as they are in the harmonic chromatic scale on p);

   (b) computing for each note n in the input and each pitch class $0 \leq p \leq 11$ a value $CNT(p, n)$ giving the number of times that p occurs within a context surrounding n that includes n, some specified number K_{pre} of notes immediately preceding n and some specified number K_{post} of notes immediately following n;

   (c) computing for each note n and each letter name l, the set of pitch classes $C(n, l) = \{p \mid S(p, n) = l\}$ (that is, the set of tonic pitch classes that would lead to n having the letter name l);

   (d) computing $N(l, n) = \sum_{p \in C(n,l)} CNT(p, n)$ for each note n and each pitch letter name l;

   (e) computing for each note n, the letter name l for which $N(l, n)$ is a maximum.

2. A method for computing the pitch names (i.e., C♯4, B♭3, etc.) of notes in a representation of music in which at least the onset time and MIDI note number (or chromatic pitch) of each note is given or can be trivially computed, comprising the steps of

   (a) computing for each pitch class $0 \leq p \leq 11$ and each note n in the input, the pitch letter name $S(p, n) \in \{A, B, C, D, E, F, G\}$, that n would have if p were the tonic at the point in the piece where n occurs (assuming that the notes are spelt as they are in the harmonic chromatic scale on p);

   (b) computing for each note n in the input and each pitch class $0 \leq p \leq 11$ a value $CNT(p, n)$ giving the number of times that p occurs within a context surrounding n that includes n, some specified number K_{pre} of notes immediately preceding n and some specified number K_{post} of notes immediately following n;

   (c) computing for each note n and each letter name l, the set of pitch classes $C(n, l) = \{p \mid S(p, n) = l\}$ (that is, the set of tonic pitch classes that would lead to n having the letter name l);

(d) computing $N(l,n) = \sum_{p \in C(n,l)} CNT(p,n)$ for each note n and each pitch letter name l;

(e) computing for each note n, the letter name l for which $N(l,n)$ is a maximum.

(f) correcting those errors in the output of the previous step in which a neighbour note or passing note is erroneously predicted to have the same letter name as either the note preceding it or the note following it, involving

i. lowering the letter name of every lower neighbour note for which the letter name predicted in step (e) is the same as that of the preceding note;

ii. raising the letter name of every upper neighbour note for which the letter name predicted in step (e) is the same as that of the preceding note;

iii. lowering the letter name of every descending passing note for which the letter name predicted in step (e) is the same as that of the preceding note;

iv. raising the letter name of every descending passing note for which the letter name predicted in step (e) is the same as that of the following note;

v. lowering the letter name of every ascending passing note for which the letter name predicted in step (e) is the same as that of the following note;

vi. raising the letter name of every ascending passing note for which the letter name predicted in step (e) is the same as that of the preceding note.

3. Computer software for computing the pitch names (i.e., C♯4, B♭3, etc.) of notes in a representation of music in which at least the onset time and MIDI note number (or chromatic pitch) of each note is given or can be trivially computed, comprising the steps of

(a) computing for each pitch class $0 \leq p \leq 11$ and each note n in the input, the pitch letter name $S(p,n) \in \{A, B, C, D, E, F, G\}$, that n would have if p were the tonic at the point in the piece where n occurs (assuming that the notes are spelt as they are in the harmonic chromatic scale on p);

(b) computing for each note n in the input and each pitch class $0 \leq p \leq 11$ a value $CNT(p,n)$ giving the number of times that p occurs within a context surrounding n that includes n, some specified number K_{pre} of notes immediately preceding n and some specified number K_{post} of notes immediately following n;

(c) computing for each note n and each letter name l, the set of pitch classes $C(n,l) = \{p \mid S(p,n) = l\}$ (that is, the set of tonic pitch classes that would lead to n having the letter name l);

(d) computing $N(l,n) = \sum_{p \in C(n,l)} CNT(p,n)$ for each note n and each pitch letter name l;

(e) computing for each note n, the letter name l for which $N(l,n)$ is a maximum.

4. Computer software for computing the pitch names (i.e., C♯4, B♭3, etc.) of notes in a representation of music in which at least the onset time and MIDI note number (or chromatic pitch) of each note is given or can be trivially computed, comprising the steps of

   (a) computing for each pitch class $0 \leq p \leq 11$ and each note n in the input, the pitch letter name $S(p,n) \in \{A,B,C,D,E,F,G\}$, that n would have if p were the tonic at the point in the piece where n occurs (assuming that the notes are spelt as they are in the harmonic chromatic scale on p);

   (b) computing for each note n in the input and each pitch class $0 \leq p \leq 11$ a value $CNT(p,n)$ giving the number of times that p occurs within a context surrounding n that includes n, some specified number K_{pre} of notes immediately preceding n and some specified number K_{post} of notes immediately following n;

   (c) computing for each note n and each letter name l, the set of pitch classes $C(n,l) = \{p \mid S(p,n) = l\}$ (that is, the set of tonic pitch classes that would lead to n having the letter name l);

   (d) computing $N(l,n) = \sum_{p \in C(n,l)} CNT(p,n)$ for each note n and each pitch letter name l;

   (e) computing for each note n, the letter name l for which $N(l,n)$ is a maximum.

   (f) correcting those errors in the output of the previous step in which a neighbour note or passing note is erroneously predicted to have the same letter name as either the note preceding it or the note following it, involving

      i. lowering the letter name of every lower neighbour note for which the letter name predicted in step (e) is the same as that of the preceding note;

      ii. raising the letter name of every upper neighbour note for which the letter name predicted in step (e) is the same as that of the preceding note;

      iii. lowering the letter name of every descending passing note for which the letter name predicted in step (e) is the same as that of the preceding note;

      iv. raising the letter name of every descending passing note for which the letter name predicted in step (e) is the same as that of the following note;

      v. lowering the letter name of every ascending passing note for which the letter name predicted in step (e) is the same as that of the following note;

      vi. raising the letter name of every ascending passing note for which the letter name predicted in step (e) is the same as that of the preceding note.

5. Computer hardware for computing the pitch names (i.e., C♯4, B♭3, etc.) of notes in a representation of music in which at least the onset time and MIDI note number (or chromatic pitch) of each note is given or can be trivially computed, comprising the steps of

   (a) computing for each pitch class $0 \leq p \leq 11$ and each note n in the input, the pitch letter name $S(p, n) \in \{A, B, C, D, E, F, G\}$, that n would have if p were the tonic at the point in the piece where n occurs (assuming that the notes are spelt as they are in the harmonic chromatic scale on p);

   (b) computing for each note n in the input and each pitch class $0 \leq p \leq 11$ a value $CNT(p, n)$ giving the number of times that p occurs within a context surrounding n that includes n, some specified number K_{pre} of notes immediately preceding n and some specified number K_{post} of notes immediately following n;

   (c) computing for each note n and each letter name l, the set of pitch classes $C(n, l) = \{p \mid S(p, n) = l\}$ (that is, the set of tonic pitch classes that would lead to n having the letter name l);

   (d) computing $N(l, n) = \sum_{p \in C(n,l)} CNT(p, n)$ for each note n and each pitch letter name l;

   (e) computing for each note n, the letter name l for which $N(l, n)$ is a maximum.

6. Computer hardware for computing the pitch names (i.e., C♯4, B♭3, etc.) of notes in a representation of music in which at least the onset time and MIDI note number (or chromatic pitch) of each note is given or can be trivially computed, comprising the steps of

   (a) computing for each pitch class $0 \leq p \leq 11$ and each note n in the input, the pitch letter name $S(p, n) \in \{A, B, C, D, E, F, G\}$, that n would have if p were the tonic at the point in the piece where n occurs (assuming that the notes are spelt as they are in the harmonic chromatic scale on p);

   (b) computing for each note n in the input and each pitch class $0 \leq p \leq 11$ a value $CNT(p, n)$ giving the number of times that p occurs within a context surrounding n that includes n, some specified number K_{pre} of notes immediately preceding n and some specified number K_{post} of notes immediately following n;

   (c) computing for each note n and each letter name l, the set of pitch classes $C(n, l) = \{p \mid S(p, n) = l\}$ (that is, the set of tonic pitch classes that would lead to n having the letter name l);

   (d) computing $N(l, n) = \sum_{p \in C(n,l)} CNT(p, n)$ for each note n and each pitch letter name l;

(e) computing for each note n, the letter name l for which $N(l, n)$ is a maximum.

(f) correcting those errors in the output of the previous step in which a neighbour note or passing note is erroneously predicted to have the same letter name as either the note preceding it or the note following it, involving

  i. lowering the letter name of every lower neighbour note for which the letter name predicted in step (e) is the same as that of the preceding note;

  ii. raising the letter name of every upper neighbour note for which the letter name predicted in step (e) is the same as that of the preceding note;

  iii. lowering the letter name of every descending passing note for which the letter name predicted in step (e) is the same as that of the preceding note;

  iv. raising the letter name of every descending passing note for which the letter name predicted in step (e) is the same as that of the following note;

  v. lowering the letter name of every ascending passing note for which the letter name predicted in step (e) is the same as that of the following note;

  vi. raising the letter name of every ascending passing note for which the letter name predicted in step (e) is the same as that of the preceding note.